

Vývoj aplikací pro Android

Android Application Development

Zadání bakalářské práce

Student: **René Matuszek**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Vývoj aplikací pro Android**
Android Application Development

Zásady pro vypracování:

Platforma Android patří mezi významné hráče na poli mobilních systémů. Cílem této práce je ilustrovat možnosti systému. Zároveň by práce měla poskytnout informace o vývoji na dané platformě, jejím specifikům, požadavkům, procesům, apod.

1. Popište specifiky platformy Android, a to jak z pohledu vývoje, tak z pohledu distribuce a nasazení aplikace.
2. Navrhněte aplikaci tak, aby byla koncepčně užitečná a bude uživatelsky použitelná. Aplikace může např. využívat dostupné API a na základě komunikace vizualizovat fotografie, apod.
3. Implementujte mobilní aplikaci, která bude ilustrovat vybrané klíčové prvky platformy.
4. Pokud to bude možné, zajistěte distribuci aplikace mezi uživatele a tento proces zhodnoťte, včetně případné uživatelské zpětné vazby.

Seznam doporučené odborné literatury:

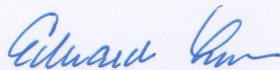
J. Friesen: Learn Java for Android Development, 2013, Apress, ISBN: 978-1430257226
Z. Mednieks: Programming Android: Java Programming for the New Generation of Mobile Devices, 2012, O'Reilly Media, ISBN: 978-1449316648

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

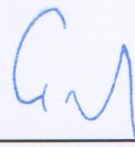
Vedoucí bakalářské práce: **Ing. Michal Radecký, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 6. května 2014



.....

Rád bych na tomto místě poděkoval všem, kteří mi pomohli s touto bakalářskou prací, vedoucímu mé bakalářské práce Ing. Michalu Radeckému, Ph.D. za jeho rady a připomínky a také bych chtěl poděkovat firmě Navmatix a jejímu jednateři Mirkovi Dolečkovi za možnost použít aplikaci, kterou jsem pro ně vyvíjel jako součást mé bakalářské práce.

Abstrakt

Cílem této práce je ilustrovat možnosti systému Android, který patří mezi významné hráče na poli mobilních systémů. Možnosti operačního systému Android jsou prezentovány na konkrétní aplikaci pro oblast Geodézie. V práci je věnován důraz na specifika vývoje pro platformu Android a možnostem distribuce aplikace k uživateli, včetně jejich zpětné vazby. Práce se dále zaměřuje na použití jednotlivých komponent systému Android, hlavně komponenty pro zobrazování map.

Klíčová slova: Android, Google, mapa, GNSS, GPS, GLONASS, referenční stanice, Top-NetLive

Abstract

The goal of this work is to show the possibilities of the Android systems which belongs to significant players on the field of mobile systems. The possibilities of the Android operating system are presented on the concrete application for geodesy field. In this work the emphasis is devoted to the developer specifics for the Android platform and to the possibilities of the application distribution to the user including the possibility of their feedback. The work is also focused on the usage of the individual components of the Android system, mainly on the components for the map projection.

Keywords: Android, Google, map, GNSS, GPS, GLONASS, permanent station, Top-NetLive

Seznam použitých zkratk a symbolů

195	– Ionospheric Noise 195 Index
ADB	– Android Debug Bridge
ADT	– Android developer tools
ANR	– Application Not Responding
APK	– Android application package file
ASP	– Active Server Pages
AVD	– Android Virtual Devices
DDMS	– Dalvik Debug Monitor Server
GLONASS	– Globalnaja navigacionnaja sputnikovaja sistema - ruský navigační systém
GNSS	– Global Navigation Satellite System
GPS	– Global Positioning System - americký navigační systém
GSM	– Globální Systém pro Mobilní komunikaci
HTTP	– Hypertext Transfer Protocol
iOS	– mobilní operační systém
JSON	– JavaScript Object Notation
KML	– Keyhole Markup Language
NTRIP	– Networked Transport of RTCM via Internet Protocol
PC	– personal computer)
PING	– Packet InterNet Groper
PNG	– Portable Network Graphics
SD	– Secure Digital - paměťová karta
SDK	– Software development kit
USB	– Universal Serial Bus
XML	– Extensible Markup Language

Obsah

1	Úvod	5
2	Specifika platformy Android	6
2.1	Úvod	6
2.2	Architektura	6
2.3	Vývojové nástroje	7
2.4	Specifika vývoje	8
2.5	Možnosti distribuce a nasazení	9
3	GNSS sítě a její rozšiřující systémy	11
3.1	Úvod	11
3.2	Základní určování polohy - kódová měření	11
3.3	Relativní určování polohy	11
3.4	Sít permanentních (referenčních) stanic	11
3.5	Vlivy ovlivňující přesnost měření	13
4	Návrh aplikace	14
4.1	Aplikace TopNET live Mobile	14
4.2	Uživatelé aplikace	14
4.3	Požadavky na aplikaci	15
4.4	Nefunkční požadavky	19
4.5	Funkce systému podle rolí	19
4.6	Uživatelské rozhraní, wireframe	20
4.7	Sekvenční a třídní diagram	20
5	Implementace aplikace	24
5.1	Použité designové vzory	24
5.2	Aktivity a fragmenty	24
5.3	Použité komponenty a knihovny	26
5.4	Implementace Google map	27
5.5	Implementace webových služeb	30
5.6	Lokalizace polohy	30
5.7	Implementace cache	30
5.8	Použití vnořeného listboxu	31
5.9	Implementační problémy	33
6	Distribuce aplikace	35
6.1	Počet instalací	35
6.2	Verze systému	35
6.3	Jazyková verze	36
6.4	Verze aplikace	36
6.5	Selhání aplikace a chyby ANR	37

7 Závěr	38
8 Reference	39

Seznam obrázků

1	Architektura převzato z http://www.androidmarket.cz	7
2	Eclipse	7
3	Emulátor	9
4	GNSS přijímač převzato z http://http://global.topcon.com/	12
5	GNSS anténa převzato z http://http://global.topcon.com/	12
6	Příklad zobrazení indexu 195 v mapě	13
7	Buffer referenčních stanic	13
8	Skyplot	14
9	Registrace	15
10	Monitoring	16
11	Ionosférický model	16
12	Buffer referenčních stanic	17
13	Nejbližší referenční stanice	17
14	Seznam stanic	18
15	Ntrip Checker	19
16	WireFrame	20
17	Sekvenční diagram	21
18	Diagram tříd Aktivita Main	22
19	Diagram tříd Aktivita Navigation	23
20	Designový vzor SplitActionBar převzato z http://developer.android.com/	24
21	Klíč pro mapy	27
22	TopNET Live Mobile na Google play	35
23	Aktuální počet instalací podle verze Android systému	35
24	Aktuální počet instalací podle jazyka uživatele	36
25	Aktuální počet instalací podle verze aplikace	36
26	Selhání aplikace a chyby ANR	37

Seznam výpisů zdrojového kódu

1	AndroidManifest	27
2	Použití mapy v kódu	28
3	Map element v layout.xml	28
4	Použití objektu marker v mapě	29
5	Použití obrázku v mapě	29
6	Layout pro ExpandableListView	31
7	Použití ExpandableListView	32

1 Úvod

Platforma Android se stává dominantním systémem pro mobilní zařízení, více než 75% všech mobilních zařízení je postaveno na systému Android.[3]

Na Android Marketu je k dispozici více než 1 000 000 aplikací [7] ale i přes toto ohromné množství různých aplikací, je v některých specifických oblastech jako např. v geoinformačních technologiích poměrně prostor pro vývoj aplikací. V oblasti GNSS síti existuje pouze několik aplikací, které většinou neobsahují možnost práce s mapou, monitoring sítě a jednoduchou registraci do sítě. Aplikace pro tuto oblast existují většinou pro platformu Microsoft Windows Mobile, což je v dnešní době již zastaralá platforma a zařízení pro tuto platformu jsou většinou velmi drahá a výkonnostně poměrně slabá, proto je snaha převést aplikace z této platformy na moderní operační systémy jako je Android.

Aplikace vyvíjena v této práci bude komplexní, bude obsahovat možnost práce s mapou, monitoring sítě a registraci do sítě. Tato aplikace bude funkčně vycházet z aplikace pro Windows Mobile TopNetLive.

2 Specifika platformy Android

2.1 Úvod

Android je otevřený operační systém založený na systému Linux pro mobilní zařízení. Z pohledu vývojáře jsou aplikace pro Android psané v jazyce Java. Společnost Google poskytuje Android SDK, které obsahuje všechny potřebné knihovny a aplikace nutné pro konverzi Java kódu na formát, který může běžet v Android zařízeních. Jako vývojové prostředí se používá Eclipse nebo příkazová řádka pro vytvoření aplikace.[3]

2.2 Architektura

Architektura systému Android je postavena na několika vrstvách obrázek 1.

Nejnižší vrstvu tvoří jádro operačního systému, které je postaveno na operačním systému Linux ve verzi 2.6. Android implementuje vlastní grafické rozhraní, nepoužívá standardní grafické uživatelské rozhraní X Window System a ani úplnou sadu GNU knihoven.

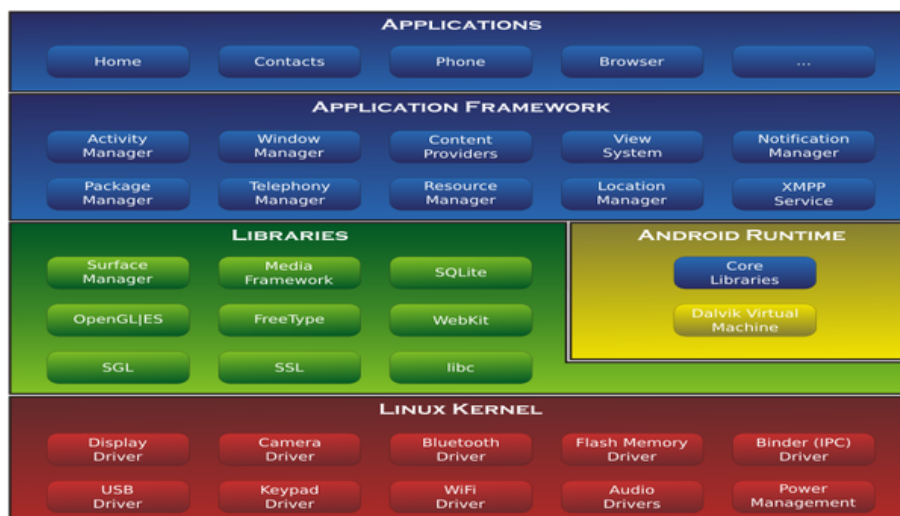
Android obsahuje hodně vlastních komponent a knihoven napsaných v jazyce C a C++ , které jsou k dispozici pod vrstvou **Android Application Framework**.

Další vrstvou je **Android Runtime**, která obsahuje virtuální stroj Dalvik, který je obdobou Java Virtual Machine.

Vrstva **Application framework** je pro vývojáře nejdůležitější. Poskytuje přístup k velkému počtu služeb, které mohou být použity přímo v aplikacích. Tyto služby mohou zpřístupňovat data v jiných aplikacích, prvky uživatelského rozhraní, upozorňovací stavový řádek, aplikace běžící na pozadí, hardware používaného zařízení a mnoho dalších služeb a funkcí. Základní sada služeb zahrnuje především:

- Sada prvků View – Tyto prvky jsou použity pro sestavení uživatelského rozhraní jako seznamy, textové pole, tlačítka, zaškrťovací políčka a jiné.
- Content providers – Umožňuje přístup k obsahu (např. kontakty) jiných aplikací.
- Resource manager – Poskytuje přístup „nekódovým“ zdrojům, jako jsou řetězce, grafika, přidané soubory.
- Notification manager – Umožňuje všem aplikacím zobrazit vlastní upozornění ve stavovém řádku.
- Activity manager – Řídí životní cyklus aplikací a poskytuje orientaci v zásobníku s aplikacemi.

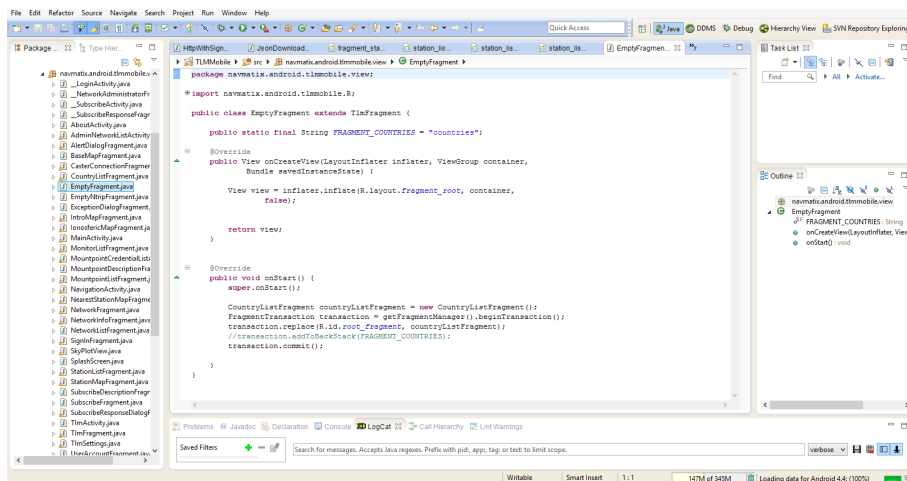
Nejvyšší vrstvu systému tvoří základní aplikace, které využívají běžní uživatelé. Může jít o aplikace předinstalované nebo dodatečně stažené z Android Marketu. Například e-mailový klient, SMS program, kalendář, mapy, prohlížeč, kontakty a další aplikace i od „třetích“ stran. [8]



Obrázek 1: Architektura převzato z <http://www.androidmarket.cz>

2.3 Vývojové nástroje

Pro vývoj na platformě Android existuje několik možností. Oficiálně podporované prostředí je Eclipse obrázek 2 do kterého je potřeba doinstalovat plugin ADT (Android developer tools), nebo si stáhnout verzi Eclipse s již nainstalovaným pluginem. Tento plugin pro Eclipse je dostupný pro většinu majoritních operačních systémů Windows, Mac OS, Linux. [9] Základní vývojové nástroje jsou obsažené v Android Software Development Kit (SDK). A tyto SDK se ještě dělí na SDK tools a Platform tools.



Obrázek 2: Eclipse

2.3.1 SDK tools

Tyto nástroje jsou nezávislé na konkrétní verzi Android systému.

- **Android SDK Manager** - aktualizuje Android SDK novými verzemi, doplňky a dokumentaci.
- **Dalvik Debug Monitor Server (ddms)** - jedná se o nástroj pro debuggování, který je integrován do Eclipse. Kromě standardních funkcí pro debuggování, umožňuje pořizovat snímek obrazovky(screenshot) zařízení nebo emulátoru, simulovat příchozí volání, sms, informace o poloze.
- **Android Virtual Devices (AVD) Manager** - pomocí tohoto nástroje je možné vytvářet, editovat a mazat virtuální zařízení pro emulátor.
- **Android Emulátor** - Emulátor operačního systému Android je obsažen v Android SDK. Umožňuje testovat vytvořené aplikace bez fyzického zařízení. Pomocí Android SDK a AVD Manageru je možné konfigurovat volbu síťového připojení, SD karty atd. a spouštět jednotlivá virtuální zařízení. [8]

2.3.2 Platform tools

Jsou k dispozici pro každou verzi Android systému.

- **Android Debug Bridge (adb)**, je nejdůležitějším nástrojem z Platform tools, pomocí něj se můžeme připojit k emulátoru nebo ke konkrétnímu zařízení přes usb port a nahrávat soubory do zařízení, instalovat balíčky apk a získávat informace o stavu zařízení.

2.4 Specifika vývoje

Vývoj pro platformu Android je v některých ohledech odlišný od vývoje pro klasické desktopové aplikace. Jedná se nejenom o rozdíly v debuggování, ladění a testování aplikace, ale i o odlišnosti v samotném programovém kódu.

2.4.1 Funkce main

Android se odlišuje od tradičního programovacího modelu, který je založen na funkci main, která je vstupním bodem programu. Systém Android nemá žádnou funkci main, ale používá více vstupních bodů programu, které umožňují spustit aplikace z různých míst, podle toho kde se uživatel v systému aktuálně nachází. [2]

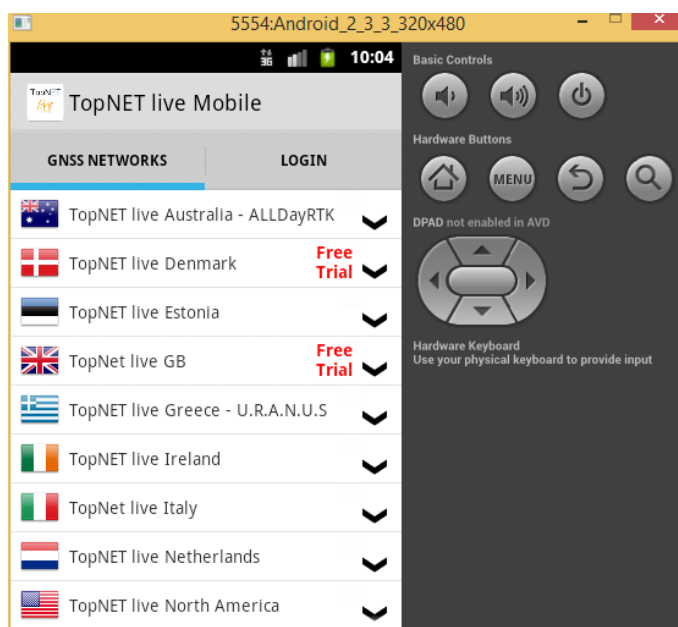
2.4.2 Kompatibilita

Při vývoji pro Android narazíme na problémy s kompatibilitou jednotlivých verzí androidu a jejich implementacích na konkrétním hardware. V současné době se aktivně

používají zařízení s verzí androidu 2.3 až 4.4. Pro vývojáře je poměrně obtížné zajistit, aby aplikace fungovala na všech verzích a různých typech zařízení stejně. Vzhledem k velkému množství zařízení pro Android a k poměrně rozdílnému vybavení zařízení, je poměrně důležité navrhnout aplikaci tak, aby rozumně běžela i na slabších přístrojích s méně pamětí.

2.4.3 Ladění a testování

Na rozdíl od desktopových aplikací, které většinou můžeme ladit na vývojovém počítači, na mobilních platformách musíme použít buď emulátor obrázek 3, nebo konkrétní přístroj. Emulátor poskytovaný v rámci Android SDK je poměrně pomalý a ne všechno v něm funguje dobře. Existuje několik lepších emulátorů od jiných firem. Nejlepší možnost je ladit aplikaci přímo na konkrétním přístroji, tato volba je také doporučovaná firmou Google. [9] Je potřeba jen umožnit debuggování aplikace přes usb v nastavení.



Obrázek 3: Emulátor

2.5 Možnosti distribuce a nasazení

Existuje několik možností distribuce aplikace k uživatelům. Primárně jsou aplikace dostupné přes službu Google Play, dříve pojmenováno Android Market. Dále je možné aplikaci distribuovat přes vlastní webové stránky, nebo přímo poslat uživateli.

2.5.1 Distribuce přes Google Play

Je vhodná, pokud je potřeba aplikaci distribuovat většímu množství uživatelů. Google Play podporuje jak bezplatnou tak i placenou distribuci aplikace. Při tomto způsobu distribuce máme k dispozici nástroje pro analýzu prodeje, např. v jakých regionech se aplikace stahuje, na jaké zařízení a mnoho dalších statistik. Pro vývojáře je podstatné, že publikováním přes Google Play se nemusí zabývat jak dostat aktualizaci nové verze aplikace k uživateli, pouze publikuje novou verzi a služba Google Play již sama řeší aktualizaci k jednotlivým uživatelům. Pro publikování přes službu Google play, je potřeba zaplatit jednorázový vývojářský poplatek.

2.5.2 Distribuce přes webovou stránku

Pokud vývojář nechce používat distribuci přes Google Play, která je placená a je zde možnost publikovat aplikace přes web. Uživatel si pak pouze stáhne aplikaci a nainstaluje. Uživatel musí mít povolenou instalaci z neznámých zdrojů. Při této distribuci se nedá použít služeb Google pro prodej a aktualizaci aplikací, toto si musí vývojář zajistit ve vlastní režii.

2.5.3 Distribuce přes email

Pro rychlé publikování verze k uživateli, stačí poslat email s apk souborem aplikace. Pokud uživatel otevře email na telefonu, Android rozpozná apk soubor a automaticky nabídne uživateli možnost nainstalovat aplikaci. Uživatel musí mít pouze v nastavení povoleno instalovat aplikace z neznámých zdrojů. Tento způsob distribuce je vhodný pro testování aplikací.

3 GNSS síť a její rozšiřující systémy

3.1 Úvod

Globální družicový polohový systém (GNSS - Global Navigation Satellite System) je služba umožňující za pomoci družic autonomní prostorové určování polohy s celosvětovým pokrytím. Uživatelé této služby používají malé elektronické rádiové přijímače, které na základě odeslaných signálů z družic umožňují vypočítat jejich polohu s přesností na desítky až jednotky metrů. [12] Ale tato přesnost je pro některé typy využití nedostatečná, proto se vytvořily systémy pro zpřesnění polohy. Existuje více metod pro určení polohy, ale v této kapitole se budeme věnovat pouze základnímu principu měření polohy a relativní metodě určování polohy.

3.2 Základní určování polohy - kódová měření

Kódová měření představují základní princip měření pomocí systému GPS. Přijímač na základě přijímaných signálů určuje dobu šíření signálu z družice k přijímači. Z tohoto času je možné spočítat jeho vynásobením rychlostí šíření radiových vln zdánlivou vzdálenost. Zdánlivou vzdáleností je výsledek měření označován proto, že je zatížen chybou hodin přijímače. Ten je vybaven jen levnými krystalovými hodinami, které jsou výrazně méně přesné, než atomové hodiny na družicích.[5]

3.3 Relativní určování polohy

Přijímače mohou rovněž být použity pro určování relativní polohy vzhledem k pevně známému bodu. Tento postup může být aplikován jak v reálném čase přímo při měření v terénu, tak i při následném zpracování v kanceláři - postprocessing. Relativní určování polohy je založeno na kódových měřeních, které určitým způsobem korigujeme. Korekce se určují pomocí referenčního přijímače (reference station), který umísťujeme na bod o přesně známých souřadnicích. Z jeho měření je možné vypočítat odchylku (chybu) přijímačem určené polohy od polohy skutečné. Zjištěné odchylky je možné přenášet jako korekce do druhého (pohyblivého) přijímače (rover station).[5]

3.4 Síť permanentních (referenčních) stanic

Jedná se o konkrétní realizaci metody relativního určování polohy. Umožňuje nám dosáhnout submetrovou přesnost v určení prostorové polohy bodu, na rozdíl od metrové přesnosti, kterou nám zajistí jeden jednofrekvenční GPS přijímač. Z tohoto důvodu se začaly po celém světě budovat sítě permanentních GNSS stanic, díky kterým uživatelům stačí si pořídit pouze jeden GNSS přijímač obrázek 4, který je schopen přijímat korekční data z těchto sítí.



Obrázek 4: GNSS přijímač převzato z <http://http://global.topcon.com/>

Permanentní síť kromě jednotlivých GNSS stanic, které neustále přijímají a zaznamenávají družicová data, obsahuje řídicí středisko, které spravuje jednotlivé stanice a umožňuje uživatelům odebírat měřená data a korekce. Anténa stanice obrázek 5



Obrázek 5: GNSS anténa převzato z <http://http://global.topcon.com/>

Největší přínos GNSS permanentních sítí je tedy ekonomický, následovaný komfortem a přesností měření. Pokud měříme v permanentní síti, dostáváme homogenní výsledky na velkém území a to ve velmi krátkém čase. V současnosti nám GNSS sítě umožňují určit svou polohu s centimetrovou přesností během několika vteřin. [11]

Na území ČR jsou v provozu tyto sítě:

- Trimble VRS Now Czech – síť provozovaná firmou Trimble
- CZEPOS – síť provozovaná Zeměměřickým úřadem, tedy státem
- TopNET – síť provozována firmou Geodis

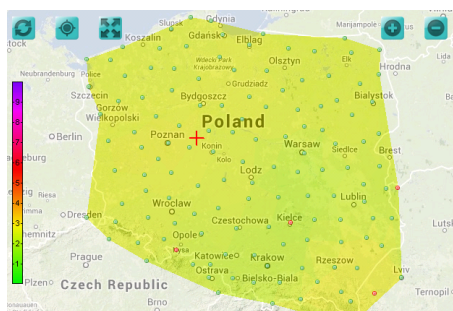
3.5 Vlivy ovlivňující přesnost měření

3.5.1 Ionospheric Noise 195 Index

Tento index vyjadřuje hodnotu intenzity ionosférické aktivity, která může ovlivňovat přesnost určování GPS polohy. Index I95 se určuje z ionosférických korekcí, které jsou počítané pro satelity nad referenčními stanicemi v určitém čase. Zobrazení indexu v mapě

Hodnoty indexu mají následující význam:

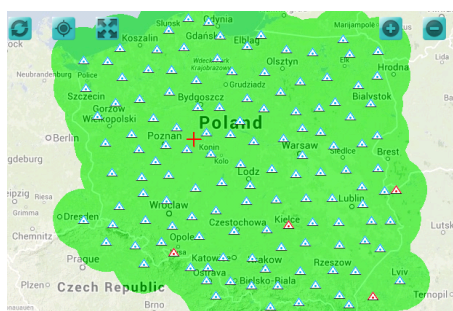
- 0-2 zanedbatelné rušení (negligible ionospheric disturbances)
- 2-4 slabé rušení (weak ionospheric disturbances)
- 4-8 silné rušení (strong ionospheric disturbances)
- 8 a více velmi silné rušení (very strong ionospheric disturbances) [10]



Obrázek 6: Příklad zobrazení indexu 195 v mapě

3.5.2 Buffer referenčních stanic

Buffer nám vymezuje pokrytí území, ve kterém se uplatňuje vliv korekcí ze sítě. Na obrázku 7 je zobrazen buffer stanic zelenou barvou, pokud provádíme měření polohy uvnitř zelené polohy, uplatňují se korekce ze sítě a měření je přesné.



Obrázek 7: Buffer referenčních stanic

4 Návrh aplikace

4.1 Aplikace TopNET live Mobile

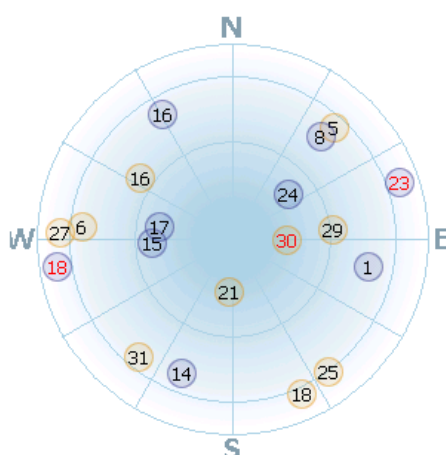
Aplikace je vyvíjena pro firmu TopCon, která provozuje síť referenčních stanic po celém světě. Firma potřebuje jednoduchou a přehlednou aplikaci jak pro nové zákazníky, kteří se chtějí registrovat do sítě, aby mohli přijímat korekce do svého gps přijímače, tak i pro administrátory k monitorování stavu sítě. Vzhledem k tomu, že měření polohy se provádí v terénu, je potřeba, aby aplikace fungovala na mobilních zařízeních. Aplikace bude pojmenována TopNET live Mobile.

4.2 Uživatelé aplikace

V sítích TopCon pracují dva druhy uživatelů, zákazníci, kteří si platí za službu sítě, že mohou přijímat korekce polohy a servisní pracovníci, administrátoři sítě, kteří sledují, zda je daná síť v pořádku.

4.2.1 Zákazník

- Základní službou sítě je poskytování korekcí polohy. Po registraci zákazníka do vybrané sítě je mu vytvořen účet a jsou mu poskytnuty přístupové údaje pro síť, které nastaví do svého GPS přijímače a může přijímat korekce ze sítě.
- Rozšířené služby sítě. Zákazník má možnost využít dalších služeb sítě, které mu poskytnou informace o stavu sítě, jaké je atmosférické rušení, jak daleko je od nejbližší referenční stanice, rozmístění satelitu (skyplot) obrázek 8 podle kterých může posoudit, jak bude měření polohy přesné



Obrázek 8: Skyplot

4.2.2 Servisní pracovník (administrátor)

Monitoruje stav sítě a zajímají ho tyto parametry:

- Počet přihlášených gps přijímačů pro odběr korekcí ze sítě (rover station)
- Stav serveru sítě, jejich dostupnost, stav a množství volného místa na disku
- Stav jednotlivých segmentů sítě, které referenční stanice jsou v provozu, které mají poruchu.

4.3 Požadavky na aplikaci

4.3.1 Registrace do sítě

Uživatel má možnost vybrat si ze seznamu sítí, do kterých se může registrovat obrázek 9. Po vyplnění údajů pro registraci, je uživateli odeslán email z informací o platbě za službu. Pokud se uživatel zaregistruje do sítě na zkoušku, je mu přímo odeslán email z informací o připojení k účtu, které pak může nastavit do svého GPS přijímače pro odběr korekce ze sítě.

GNSS NETWORKS	LOGIN
TopNET live Australia - ALLDayRTK	▼
TopNET live Denmark	Free Trial ▼
TopNET live Estonia	▼
TopNET live GB	Free Trial ▼
TopNET live Greece - U.R.A.N.U.S	▼
TopNET live Ireland	▼
TopNET live Italy	▼
TopNET live Netherlands	▼
TopNET live North America	▼
TopNET live Poland	▼
TopNET live Sweden	▼

First name
Last name
Company
Street
City
Zip code
State/Province
Country
E-mail
Phone

Obrázek 9: Registrace

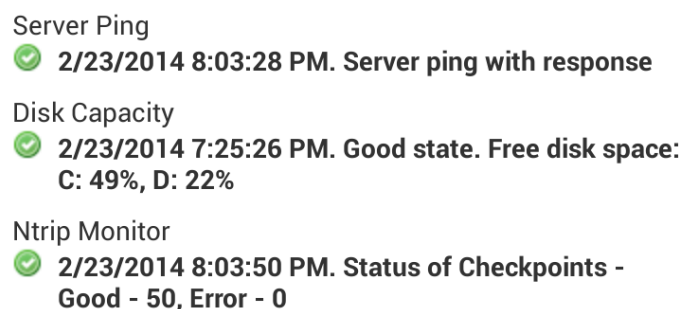
4.3.2 Výpis informací o účtu

Poskytne uživateli přehled o jeho účtu, registračních údajích, expiraci účtu. Jsou zde také statistiky o množství přenesených korekčních dat a celkové době připojení k síti.

4.3.3 Monitoring sítě

Zobrazuje informace o stavu sítě pro jednotlivé servery obrázek 10.

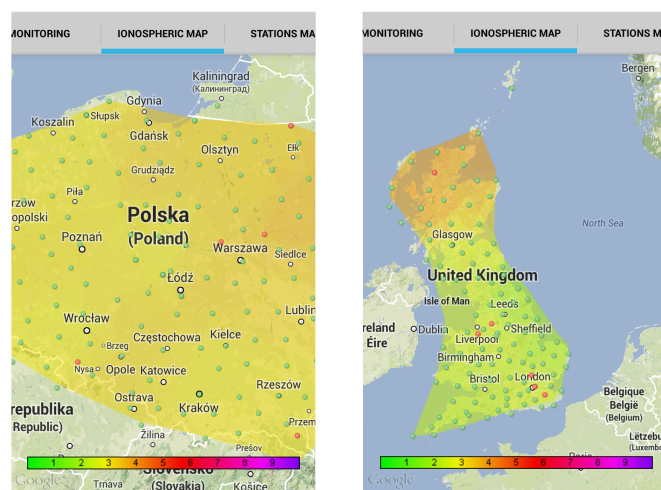
- dostupnost serveru (ping)
- Volné místo na disku serveru
- Ntrip monitor



Obrázek 10: Monitoring

4.3.4 Zobrazení ionosférického modelu

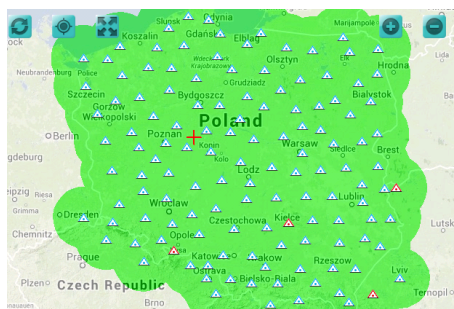
Uživatel má možnost zobrazit aktuální ionosférický index 195 do mapy sítě stanic. V mapě se budou zobrazovat i jednotlivé referenční stanice včetně jejich stavu (online, off-line). Stavby budou rozlišené ikonou stanice. V mapě bude také zobrazená aktuální poloha uživatele obrázek 11.



Obrázek 11: Ionosférický model

4.3.5 Zobrazení bufferu referenčních stanic

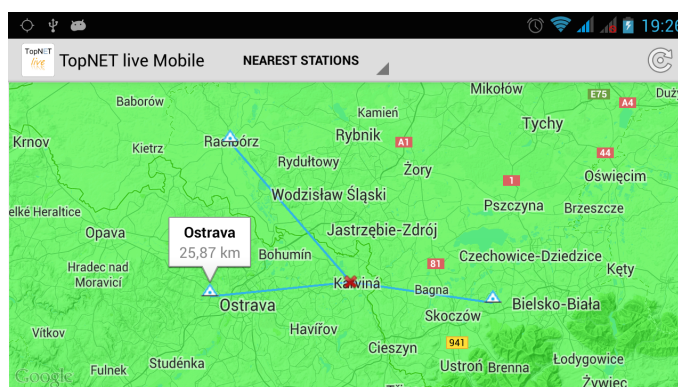
Aplikace musí zobrazovat buffer referenčních stanic do mapy, včetně jednotlivých referenčních stanic a jejich stavu (online, off-line). Stav budou rozlišené ikonkou stanice. V mapě bude také zobrazená aktuální poloha uživatele obrázek 12.



Obrázek 12: Buffer referenčních stanic

4.3.6 Zobrazení nejbližších referenčních stanic








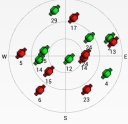







V aplikaci půjdou v mapě zobrazit referenční stanice, ke kterým je uživatel nejbliže. Budou se zobrazovat pouze 3. nejbližší stanice. V mapě bude také zobrazen buffer sítě a aktuální poloha uživatele obrázek 13.



Obrázek 13: Nejbližší referenční stanice

4.3.7 Výpis seznamu referenčních stanic

Uživatel může procházet seznam referenčních stanic sítě. U referenční stanice zobrazíme její název, stát - ve kterém se stanice nachází, stav stanice (online, off-line) a počet viditelných satelitů nad stanicí GPS a GLONASS. Po výběru referenční stanice zobrazíme rozmístění satelitů nad referenční stanicí obrázek 14.

TATIONS MAP	STATIONS LIST	NEAREST STATI	TATIONS MAP	STATIONS LIST	NEAREST STATI
	Barwice 9/10 (GPS/GLD)	▼		ABEP 11/8 (GPS/GLD)	▼
	Belchatow 8/8 (GPS/GLD)	▼		ADAR 11/9 (GPS/GLD)	▼
	Biale Blota 10/10 (GPS/GLD)	▼		ALDB 10/8 (GPS/GLD)	▲
	Bialopole 9/9 (GPS/GLD)	▼	Lat: 52,15 Long: 1,60 Alt: 56,90 		
	Bialystok 8/9 (GPS/GLD)	▼		AMER 9/8 (GPS/GLD)	▼
	Boleslawiec 9/9 (GPS/GLD)	▼		ANLX 10/9 (GPS/GLD)	▼
	Braniewo 10/10 (GPS/GLD)	▼			
	Branszczyk 0/0 (GPS/GLD)	▼			
	Brodnica 9/9 (GPS/GLD)	▼			

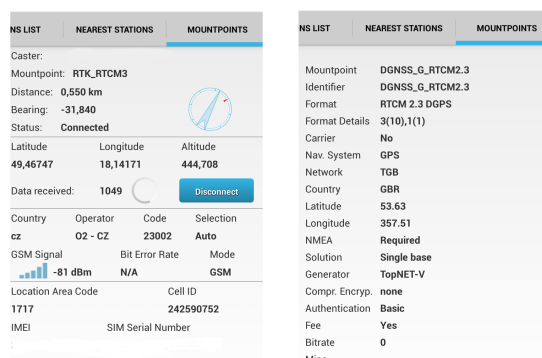
Obrázek 14: Seznam stanic

4.3.8 Ntrip Checker

Ntrip Checker slouží k ověření, že komunikace s GNSS sítí přes internet je v pořádku. Tato komunikace je založena na NTRIP protokolu, který je postaven na protokolu HTTP. Uživatel má možnost zobrazit seznam http serveru (NtripCaster), ke kterým se může připojit a pro každý NtripCaster vypsat informace obrázek 15. Po výběru serveru, zobrazíme tyto informace:

- hodnotu síly GSM signálu
- aktuální polohu uživatele
- vzdálenost k nejbližší referenční stanici
- kompas, který zobrazuje směr k nejbližší referenční stanici

Uživatel může spustit test pro vybraný NtripCaster a pak zobrazujeme množství přenesených testovacích dat.



Obrázek 15: Ntrip Checker

4.4 Nefunkční požadavky

- Aplikace musí běžet minimálně na verzi androidu 2.3
- Aplikace musí zobrazovat data v reálném čase.
- Aplikace musí být rozšiřitelná přidáním dalších modulů.
- Aplikace poskytuje citlivé údaje, přístup k nim musí být autorizován.

4.5 Funkce systému podle rolí

• Neregistrovaný, anonymní uživatel

Má možnost registrovat se do konkrétní sítě, kterou si vybere. Sítě jsou členěny dle zemí. Uživatel má možnost provést registraci na zkušební dobu - v tomto případě mu přijde email z přístupovými údaji do sítě a může ji po omezenou dobu používat. Dále se může registrovat přímo, pak mu v emailu přijdou informace o platbě apod. Pomocí přihlašovacích údajů do sítě se může přihlásit do aplikace.

• Registrovaný uživatel

Tento uživatel bude mít možnost kontroly svého uživatelského účtu, přihlašovacích údajů, platby apod. Uživatel bude mít možnost vidět základní přehled o síti, včetně informací o stavu sítě, kolik je v provozu referenčních stanic a jaké je případně ionosférické rušení. Dále bude vidět ionosférickou mapu pro danou síť a mapu pokrytí sítě (buffer sítě) včetně umístění referenčních stanic a v jakém jsou stavu. Dále budeme vypisovat seznam jednotlivých referenčních stanic včetně informací kolik je nad ní v současné době viditelných satelitu GPS, GLONASS a případně po výběru referenční stanice se objeví rozmístění satelitů graficky. Poslední funkcí systému bude zobrazení nejbližších referenčních stanic k aktuální poloze a informací o vzdálenostech k těmto stanicím.

- **Administrátor sítě**

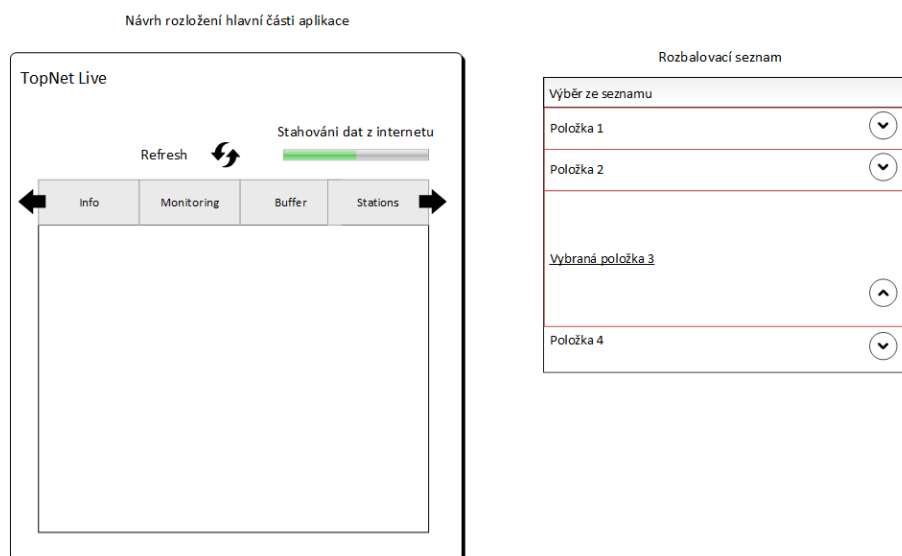
Bude mít všechny možnosti jako registrovaný uživatel, navíc pak uvidí v přehledu sítě počet uživatelů a také počet přihlášených gps přijímačů, které odebírají korekce ze sítě - provádí měření. Ještě uvidí položku monitoring, kde jsou informace o serverech sítě, zda jsou dostupné, v pořádku, nedochází místo na disku apod.

- **Administrátor**

Má stejné možnosti jako administrátor sítě, ale má navíc přístup do všech sítí, může provést výběr sítě, ve které bude pracovat.

4.6 Uživatelské rozhraní, wireframe

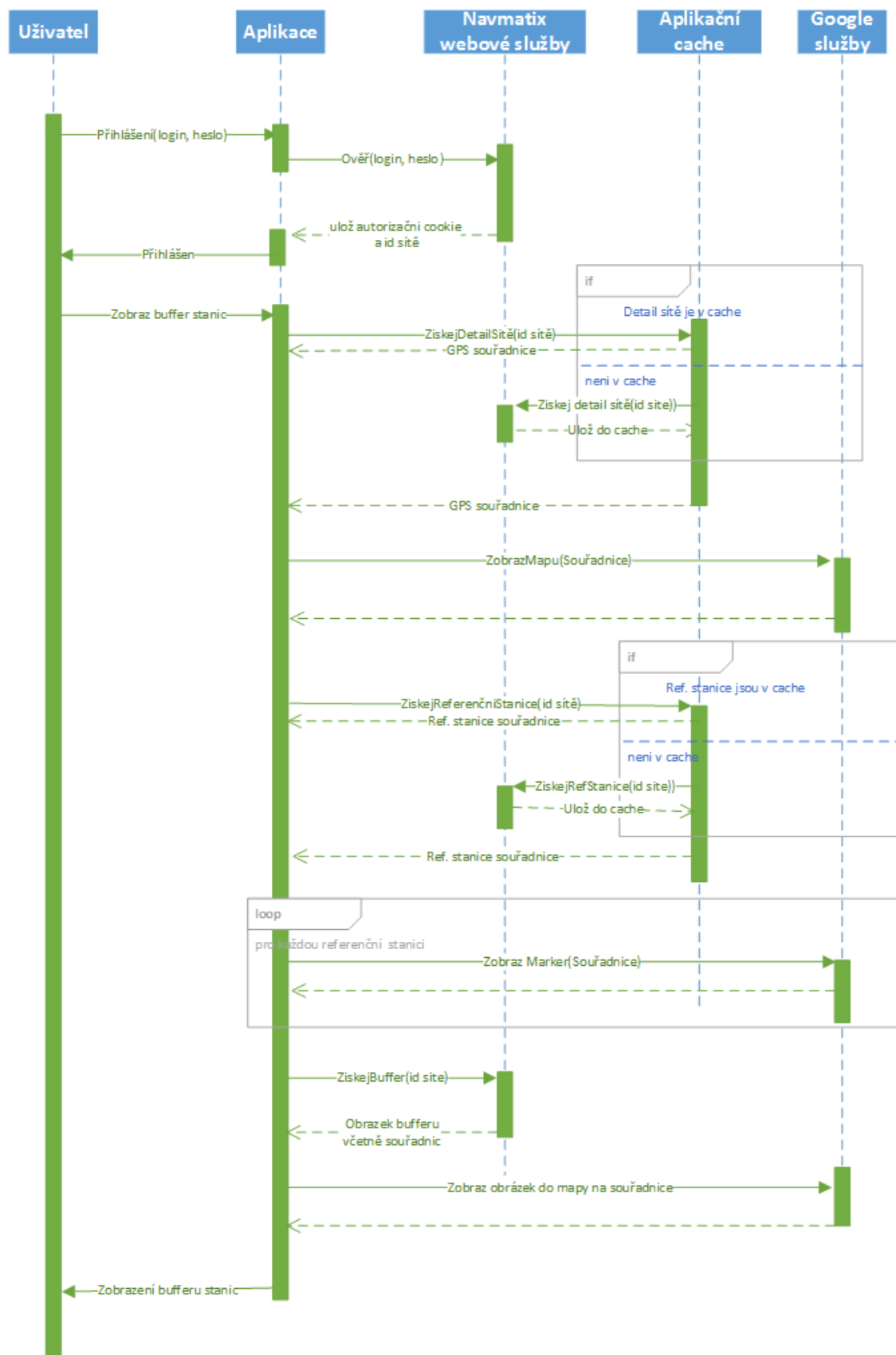
Uživatelské rozhraní je navrženo s ohledem na jednoduchost a přehlednost ovládaní. Na schématickém zobrazení uživatelského rozhraní 16 jsou dvě obrazovky aplikace. Na prvním obrázku je návrh navigace v aplikaci pomocí záložek, které uživateli umožní rychlé přepínání mezi jednotlivými moduly aplikace. Na druhém obrázku je rozevírací seznam.



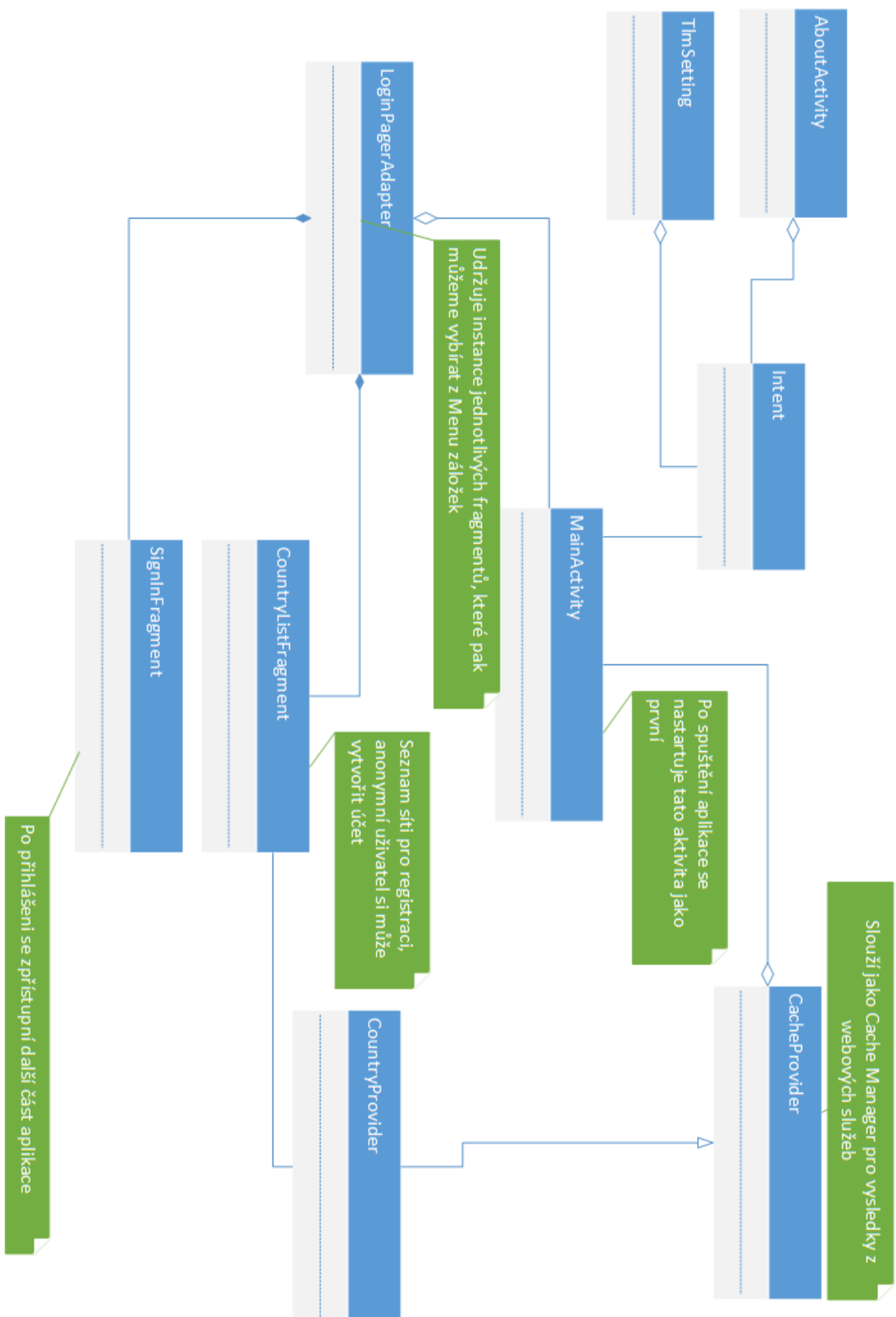
Obrázek 16: WireFrame

4.7 Sekvenční a třídí diagram

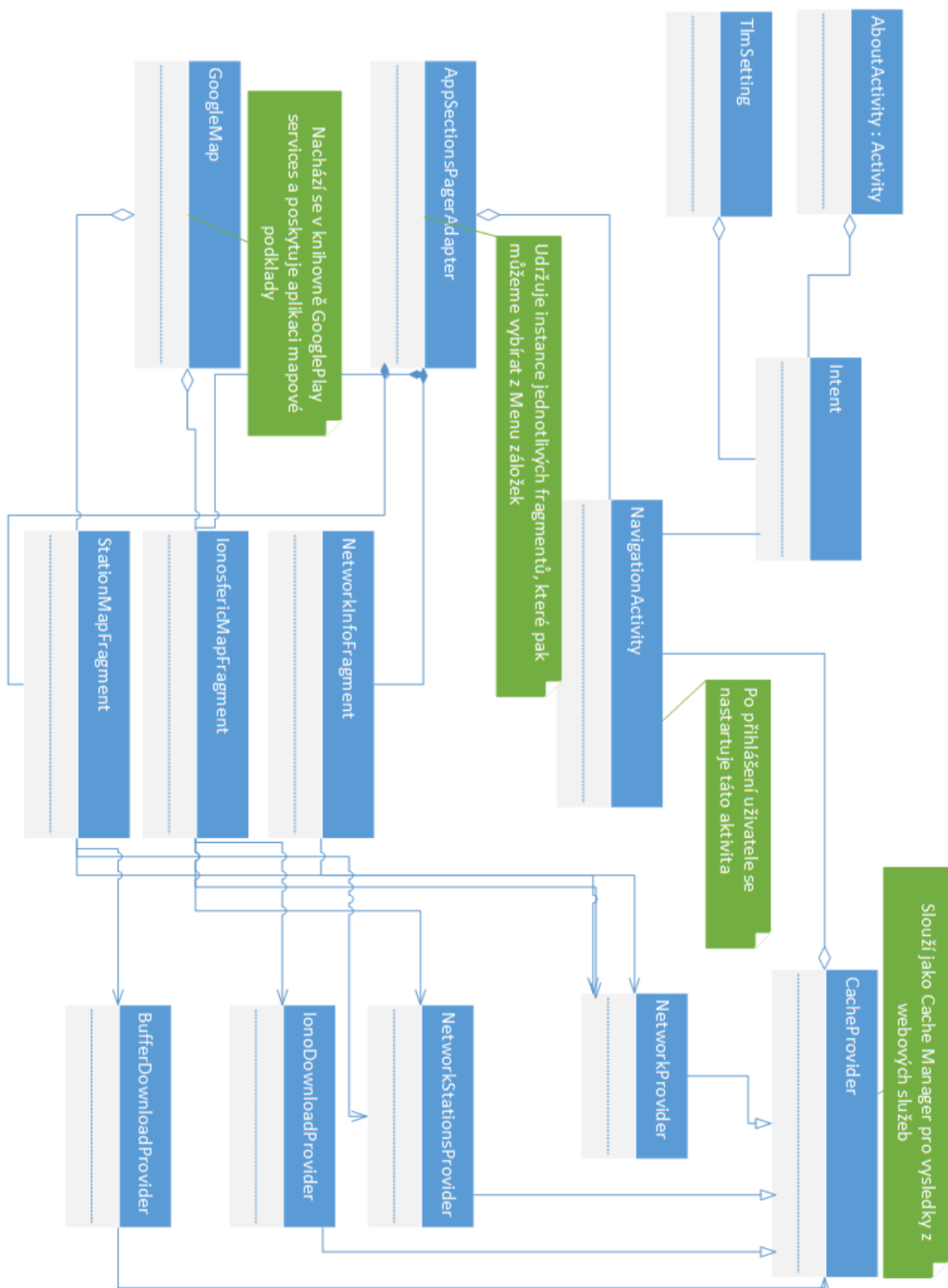
Základní funkčnost aplikace je schématicky znázorněna pomocí sekvenčního diagramu pro modul buffer stanic 17. Je zde hlavně vyznačena komunikace mezi aplikací a jednotlivými webovými službami a jejich ukládání do cache přes aplikační cache manager. Dále jsou zde zobrazeny zjednodušené třídí diagramy pro obě části aplikace, jak pro anonymního uživatele diagram 18 tak i pro uživatele přihlášeného do aplikace diagram 19.



Obrázek 17: Sekvenční diagram



Obrázek 18: Diagram tříd Aktivita Main

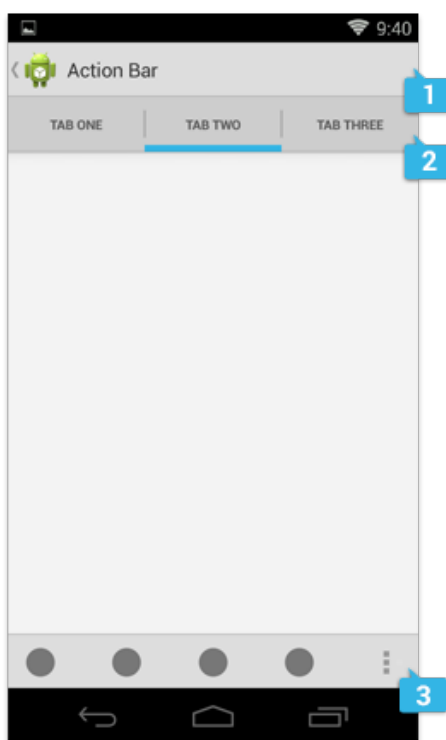


Obrázek 19: Diagram tříd Aktivita Navigation

5 Implementace aplikace

5.1 Použité designové vzory

Vzhledem k velkému množství různých obrazovek, se kterými se bude v aplikaci pracovat, bylo vhodné použít designový vzor, který společnost Google označuje jako Split Actions Bars obrázek 20, v zásadě se jedná o lištu záložek, která se dá posouvat, je vhodná pokud potřebujeme přepínat mezi více obrazovkami. Protože standardní action bar nefunguje dobře na starších verzích systému Android, byla zvolena komponenta ActionSherlokBar - která má kompatibilní api jako standardní action bar, ale má více možností a je kompatibilní i s verzí systému Android 2.3



Obrázek 20: Designový vzor SplitActionBar převzato z <http://developer.android.com/>

5.2 Aktivity a fragmenty

Aplikace používá jako hlavní menu - záložky (Actions Bars), které pracují s fragmenty, díky tomu jsou v aplikaci pouze dvě hlavní aktivity, které pracují s jednotlivými fragmenty, do kterých jsou rozdělené jednotlivé funkční bloky.

Aplikace je logicky rozdělená na dva samostatné celky.

- část pro anonymního uživatele, která obsahuje záložku registrace do sítě a záložku přihlášení. Tato část je postavena na MainActivity, která vytváří dvě záložky -

přihlášení do aplikace a registraci do vybrané sítě. V případě úspěšného přihlášení se spustí `NavigationActivity`, a přepne uživatele do druhé části aplikace pro monitoring sítě. Další částí je záložka s registrací do sítě a ta pracuje s těmito fragmenty:

- **CountryListFragment**

V tomto fragmentu implementujeme `ExpandableListView`, kde zobrazujeme seznam dostupných sítí. Po výběru konkrétní sítě se zobrazí možnost výpisu informace o síti a možnost registrace do sítě, u některých sítí je navíc možnost registrace. V tomto fragmentu vytváříme `CountryCursorAdapter`, který je potomkem `CursorTreeAdapter`, pomoci této třídy plníme daty `ExpandableListView`, data dostaneme z databáze, kde byly dříve uloženy pomocí webových služeb.

- **SubscribeDescriptionFragment**

Tady se zobrazí jen popis informace o síti, podmínky platby za využívání sítě apod.

- **SubscribeFragment**

Zobrazíme registrační formulář, který se pak vyplněný odešle na server. Uživatel by měl přijít potvrzující email na adresu, kterou vyplnil ve formuláři. Pokud se registruje pro trial použití, přijdou mu rovnou informace pro připojení k síti.

- část pro přihlášeného uživatele obsahuje záložky pro monitoring sítě. V této části pracujeme v `NavigationActivity`, ve které vytvoříme záložky pro monitoring sítě. Každá záložka je reprezentována samostatným fragmentem

- **NetworkInfoFragment**

Vypisuje informace o síti, výpis je závislý na roli přihlášeného uživatele

- **IonosfericMapFragment**

Zobrazuje ionosférický model aplikace v Google map komponentě. V mapě zobrazíme referenční stanice jako objekty typu `Marker`. Ikonu pro `Marker` přiřazujeme dle stavu stanice. Pro zobrazení ionosféry používáme dva soubory, jeden obrázek, který je generován na serveru a druhý textový, který obsahuje souřadnice, kam ho umístit do mapy. Vkládáme ho jako objekt typu `GroundOverlayOptions`.

- **StationMapFragment**

Zobrazuje buffer sítě (pokrytí sítě) včetně referenčních stanic. Realizace je obdobná jako u fragmentu `IonosfericMap`

- **StationListFragment**

V `ExpandableListView` zobrazujeme seznam referenčních stanic. Po výběru stanice se zobrazí rozmístění satelitů nad touto stanicí. Pomoci třídy `StationCursorAdapter`, který je potomkem `CursorTreeAdapter` plníme `ExpandableListView` daty z databáze, která byla naplněna pomocí webových služeb.

- **NearestStationMapFragment**

Zobrazuje buffer sítě, včetně 3 nejbližších stanic. Současnou polohu získáme pomocí LocationManager a GPS providera a zašleme do webové služby, která nám vrátí 3. nejbližší stanice a jejich vzdálenosti a ty zobrazíme jako objekty Marker do mapy.

- **MonitorListFragment**

Zobrazuje informace o stavu jednotlivých serveru, na kterých je provozovaná síť, získáme informace o dostupnosti serveru, volné kapacitě disku atd. Tyto informace jsou dostupné pouze roli administrátor.

Dále se v aplikaci používají aktivity pro nastavení aplikace TlmSettings která je potomkem PreferenceActivity - standardní třída pro nastavení parametrů aplikace a AboutActivity - pouze pro informace o verzi programu. Pro administrátora je k dispozici ještě aktivita AdminNetworkListActivity, která mu umožní přepnutí mezi jednotlivými sítěmi.

5.3 Použité komponenty a knihovny

- **Google play services**

Nejdůležitější komponentou v aplikaci jsou Google map, které jsou obsaženy v knihovně Google play services. Jedná se o standardní komponentu od společnosti Google, která se dá stáhnout přes SDK Manager. Jedná se vlastně o klienta na straně telefonu, který se připojuje k serverům, které provozuje Google s různými službami. V tomto případě využíváme pouze mapy. Přes vývojářskou konzoli, je třeba ještě aplikaci zaregistrovat k této službě a vložit vygenerovaný klíč do manifestu aplikace. V aplikaci je ještě Google play services použita pro získání aktuální polohy uživatele - kapitola lokalizace polohy.

- **Action bar sherlock**

Tato komponenta je použita pro implementaci designového vzoru action bar pattern. Google používá vlastní třídu ActionBar, ale tato je až ve verzi Android 3.0 a vyšší. Pro nižší verzi je dostupná přes support library. Externí komponenta ActionBarSherlock, má stejné api, jako komponenta Googlu a je funkční i v nižších verzích androidu. Komponenta je ke stažení na <http://actionbarsherlock.com/download.html>

- **Ekito, Simple-KML**

Slouží pro zpracování kml dat. Kml soubory obsahují popisy geografických objektů, které pak můžeme vykreslit do mapy. Bohužel Google nemá vlastní komponentu, pro tuto funkci, tak byla použita volně dostupná a společnosti Google doporučená komponenta Simple-KML. Knihovna je publikována pod Apache Software License 2.0. Knihovna je ke stažení na adrese <https://github.com/Ekito/Simple-KML>. Bohužel později se projevilo, že daná komponenta má výkonnostní problémy a bylo použito jiné řešení viz. kapitola implementační problémy.

5.4 Implementace Google map

5.4.1 Registrace

Pro použití Google map v aplikaci pro systém Android musíme aplikaci zaregistrovat. Ve vývojové konzoli na <https://console.developers.google.com/> vytvoříme projekt, ke kterému vybereme položku Google Maps Android API v2, kterou povolíme a vložíme klíč, kterým je podepsána aplikace pro publikaci obrázků 21. Můžeme použít i standardní klíč pro vývoj. Vývojová konzola nám vygeneruje sériové číslo, které je potřeba vložit do manifestu aplikace, pod element meta-data.

Pro správnou funkčnost map je potřeba dát aplikaci právo přístupu na internet.

```
...
...
<uses-permission android:name="android.permission.INTERNET" />
...
...
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"/>
```

Výpis 1: AndroidManifest

Create an Android key and configure allowed Android applications

This key can be deployed in your Android application.

API requests are sent directly to Google from your client Android device. Google verifies that each request originates from an Android application that matches one of the certificate SHA1 fingerprints and package names listed below. You can discover the SHA1 fingerprint of your developer certificate using the following command:

```
keytool -list -v -keystore mystore.keystore
```

[Learn more](#)

Accept requests from an Android application with one of the certificate fingerprints and package names listed below

One SHA1 certificate fingerprint and package name (separated by a semicolon) per line. Example:

```
45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A0:E0;com.example
```

Create

Cancel

Obrázek 21: Klíč pro mapy

5.4.2 Použití mapy v kódu

Použití mapy je poměrně jednoduché a intuitivní. Google map komponentu vložíme do layoutu aplikace jako fragment a pak použijeme v kódu. Můžeme nastavit typ mapy, souřadnice středu mapy a základní přiblížení v metodě `moveCamera`.

```

public class BaseMapFragment extends TlmFragment
{
    protected GoogleMap map;

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        ...
        ...

        // Vyhledame komponentu GoogleMap v layout.xml
        map = ((SupportMapFragment) getFragmentManager().findFragmentById(R.id.station_map)).
            getMap();

        if (map == null) {
            Log.d("", "Map_wasnt_loaded_properly");
            return rootView;
        } else {
            Log.i("", "Map_loaded_fine");

            // Nastavime typ zobrazovane mapy
            map.setMapType(GoogleMap.MAP_TYPE_TERRAIN);

            // Nastavime parametry, v tomto pripade zakazeme ikonky pro zoom mapy
            map.getUiSettings().setZoomControlsEnabled(false);

            // Nastavime pozici mapy a prvotni zoom
            map.moveCamera(CameraUpdateFactory.newLatLngZoom(fullNetwork.
                getMapCenter(), (float)fullNetwork.getMapFocus()));
        }
    }
}

```

Výpis 2: Použití mapy v kódu

```

<fragment
    android:id="@+id/station_map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    class="com.google.android.gms.maps.SupportMapFragment" />

```

Výpis 3: Map element v layout.xml

5.4.3 Použití objektu marker v mapě pro zobrazení referenčních stanic

Marker vložíme do mapy metodou `map.addMarker` ve které uvedeme souřadnice bodu, obrázek pro marker a případně popis, který se po kliknutí na marker zobrazí.

```

public class BaseMapFragment extends TlmFragment
{

    protected GoogleMap map;

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        ...
        ...

        // Vyhledame komponentu GoogleMap v layout.xml
        map = ((SupportMapFragment) getFragmentManager().findFragmentById(R.id.station_map)).
            getMap();

        ..

        map.addMarker(new MarkerOptions()
            .title ("Current_position")
            .position (currentPosition)    // Zadame souradnice v mape
            .flat (true)
            .anchor(0.5f, 0.5f)
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.red_cross)) //Ikona
        );

    }

```

Výpis 4: Použití objektu marker v mapě

5.4.4 Překrytí mapy obrázkem pro zobrazení bufferu stanic

Pro překrytí mapy obrázkem použijeme metodu `addGroundOverlay` do které vložíme objekt typu `GroundOverlayOptions` ve kterém definujeme samotný obrázek a souřadnice kam ho v mapě umístíme. Je nutno nastavit transparentnost překrytí.

```

public class BaseMapFragment extends TlmFragment
{

    protected GoogleMap map;

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        ...
        ...

        // Vyhledame komponentu GoogleMap v layout.xml
        map = ((SupportMapFragment) getFragmentManager().findFragmentById(R.id.station_map)).
            getMap();

        ..

```

```

GroundOverlayOptions newarkMap = new GroundOverlayOptions()
    .image(BitmapDescriptorFactory.fromFile(TlmMobile.STATION.BUFFER.PNG_FILE)) //
        Prekryvající obrazek
    .transparency(0.5f)
    .positionFromBounds(coordinateBounds);

map.addGroundOverlay(newarkMap);

}

}

```

Výpis 5: Použití obrázku v mapě

5.5 Implementace webových služeb

Webové služby jsou založené na komunikaci přes json. Webové služby jsou jak pro anonymní uživatele, nevyžadují autorizaci, tak i pro uživatele, kteří mají vlastní účet a Ti se musí do aplikace přihlásit. Samotná autorizace je na serveru postavená na technologii ASP.NET a autorizační cookie. Samotné přihlášení se provede pomocí služby SignIn, a pokud je úspěšné, pak se vytvoří cookie. Oproti dalším službám se již neautorizujeme, posíláme pouze tuto cookie. Samotná komunikace je realizována přes *java.net.HttpURLConnection*. Pro webové služby používáme metody GET i POST. Pro práci s cookie, kde potřebujeme uložit autorizační cookie se používá *java.net.CookieManager*. Pro práci s json objekty je použita třída *org.json.JSONObject*

5.6 Lokalizace polohy

Pro zobrazení aktuální polohy v mapě jsou použity Location services od společnosti Google. S touto službou pracujeme přes objekt *LocationClient*, který je obsažen v knihovně Google play services. V aplikaci jsme nastavili právo pro přístup jak k přesné poloze přes GPS, tak i pro méně přesnou polohu přes síťového poskytovatele. Pro tuto aplikaci je plně postačující určování polohy přes síťového poskytovatele, danou polohu získáme mnohem rychleji než z gps přijímače. Před inicializací Location services ověřujeme, zda je tato služba, resp. její knihovna na cílovém zařízení a je ve správné verzi. Pokud ne nabídneme uživateli řešení (instalaci knihovny, případně její aktualizaci).

5.7 Implementace cache

Vzhledem k tomu, že volání webových služeb může být relativně pomalé, implementujeme v aplikaci cache. Položky v cache jsou rozdělené na dvě části - dlouhodobá cache expirace nastavena na 24 hodin, je určena pro webové služby, které jsou statické jako seznam sítí pro registraci, seznam sítí pro administrátory. Jedna se o položky, které se mění zřídka. Druhá část cache je krátkodobá a každá webová služba má vlastní dobu

expirace. Doba expirace se získá z webové služby Exspirace cache, která se zavolá po spuštění aplikace. Jedná se o položky, které jsou dynamické a doba uchování položky v cache je v řádu minut. Aplikace nevolá přímo webové služby, ale pracuje s cache, která je plněna transparentně pouze při expiraci. Jako úložiště pro cache používáme lokální databázi sqllite, která je standardní komponentou systému Android.

5.8 Použití vnořeného listboxu

V aplikaci se často používá třída *ExpandableListView* k implementaci vnořeného listboxu. Je zde uveden symbolický výpis kódu s implementací této komponenty. Pro názornost jsou v něm uvedeny jenom klíčové prvky. Pokud používáme *ExpandableListView* je třeba ho nadefinovat v xml souboru pro layout aplikace, kde je reprezentován elementem *ExpandableListView*. Kromě standardních formátovacích atributu jako pro ostatní view, je potřeba doplnit *android:groupIndicator* pro zobrazení ikon označujících, zda je seznam rozbalen.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <ExpandableListView
        android:id="@+id/ station_list_list "
        android:layout_width=" fill_parent "
        android:layout_height=" fill_parent "
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:groupIndicator="@drawable/group_indicator"
    />

</RelativeLayout>
```

Výpis 6: Layout pro ExpandableListView

Komponenta *ExpandableListView* je navržena pro práci s objektem typu *CursorTreeAdapter*, který ji poskytuje data. V tomto případě jsme vytvořili objekt *StationCursorAdapter*, který je potomkem *CursorTreeAdapter* a implementovali tyto základní metody:

- *View newGroupView(Context context, Cursor cursor, boolean isExpanded, ViewGroup parent)* V této metodě vytvoříme View, které reprezentuje položku seznamu, která není rozbalená
- *View newChildView(Context context, Cursor cursor, boolean isLastChild, ViewGroup parent)* V této metodě vytvoříme View, které reprezentuje položku seznamu, která je vybraná (rozbalená)
- *bindGroupView(View view, Context context, Cursor cursor, boolean isExpanded)* V této metodě naplníme položky view daty z kurzoru pro položky, které nejsou rozbalené

- *bindChildView(View view, Context context, Cursor cursor, boolean isLastChild)* V této metodě naplníme položky view daty z kurzoru pro položky, které jsou vybrané

```

public class StationListFragment extends TlmFragment implements BaseProviderListener,
    NetworkActivityIndicator, OnItemClickListener {

    class StationCursorAdapter extends CursorTreeAdapter {

        public StationCursorAdapter(Context context, Cursor c) {
            super(c, context);
            skyplotViews=new Hashtable<Long,SkyPlotView>();
        }

        @Override
        protected void bindChildView(View view, Context context, Cursor cursor,
            boolean isLastChild) {

            Long stationId = cursor.getLong(cursor.getColumnIndex(NetworkStationConst.
                STATION_ID));

            TextView textViewLongitude = (TextView)view.findViewById(R.id.sky_plot_longitude);

            .....

        }

        @Override
        protected void bindGroupView(View view, Context context, Cursor cursor,
            boolean isExpanded) {

            long stationId = cursor.getLong(cursor.getColumnIndex(NetworkStationConst.
                STATION_ID));
            String name = cursor.getString(cursor.getColumnIndex(NetworkStationConst.NAME));

            holder.name.setText(name);
            holder.satCount.setText(String.format("%s/%s_(GPS/GLO)", gpsCount, glonasCount));

            .....

        }

        @Override
        protected Cursor getChildrenCursor(Cursor groupCursor) {
            .....
            MatrixCursor mc = new MatrixCursor(new String[] { "_id", NetworkStationConst.
                STATION_ID, NetworkStationConst.Coordinate.LONGITUDE, NetworkStationConst.
                Coordinate.LATITUDE, NetworkStationConst.Coordinate.ALTITUDE});
            mc.addRow(new Object[] { "1", stationId, longitude, latitude, altitude });
            return mc;
        }
    }

```

```

@Override
protected View newChildView(Context context, Cursor cursor,
    boolean isLastChild, ViewGroup parent) {
    LayoutInflater inflater = LayoutInflater.from(context);
    View view = inflater.inflate(R.layout.station_list_skyplot, parent,
        false);

    return view;
}

@Override
protected View newGroupView(Context context, Cursor cursor,
    boolean isExpanded, ViewGroup parent) {
    LayoutInflater inflater = LayoutInflater.from(context);
    View view = inflater.inflate(R.layout.station_list_item_2row, parent,
        false);

    return view;
}
}
}

```

Výpis 7: Použití ExpandableListView

5.9 Implementační problémy

5.9.1 Zpracování kml souboru

Největší implementační problém nastal při zpracování kml souboru, který se stahuje z webových služeb pro ionosférický model a buffer. Zpracování cca 600 objektů obsažených v běžném kml souboru trvalo na dvoujádrovém telefonu s procesorem o rychlosti 1.5 GHz cca 2 sekundy, což bylo ještě přijatelné. V reálném provozu vznikl problém na pomalejších zařízeních, kde zpracování někdy trvalo i 20. sekund, což již nebylo použitelné. Jedno řešení bylo vytvořit kml soubor z méně detaily, pak to běželo rychleji, ale ve výsledku to graficky nebylo pěkné, nebo převádět kml soubor přímo na serveru do formátu png a poslat na klienta se souřadnicemi, kam na mapě umístit tento obrázek. Nakonec se použila druhá možnost, která je nezávislá na výkonu telefonu tedy je vždy rychlá. Bohužel, když se zavedl ionosférický model pro Americkou síť, která pokrývá celé území Ameriky, objevili se v obrázkových souborech deformace. Vznikl problém se zakulacením země, které se u menších evropských sítí neprojevilo, ale na velkém území je vidět nepřesnosti. V další verzi aplikace bude potřeba provést rozdělení a vytvoření kml souboru do více obrázku a ty pak vložit do mapy.

5.9.2 Emulátor, zobrazení map

Při vývoji aplikace v emulátoru, se zjistilo, že ve standartním emulátoru, který je poskytnuty v rámci SDK pro Android nejde spustit komponenta pro mapy. Problém byl v tom, že emulátor nepodporuje služby Google play seivces, které používá komponenta pro práci z mapou. Existují určité postupy jak zprovoznit mapy v emulátoru, ale v tomto případě se mi nepodařily mapy pod emulátorem zprovoznit. Vývoj dále probíhal přímo na fyzickém zařízení.


6 Distribuce aplikace

Distribuce aplikace je provedena přes Google play. Díky tomu je možné sledovat statistiku stažení aplikaci, na jakých zařízeních se provozuje. Uživatelé mají možnost ohodnotit aplikaci, případně pokud aplikace nefunguje správně odeslat tuto informaci k vývojáři aplikace. Pro vývojáře jsou statistiky z Google play přístupné přes vývojářskou konzoli. Aplikace je publikována pod názvem **TopNET Live Mobile**. Poslední verze aplikace je označena jako 3.1.0 Statistiky z Google play jsou pořízeny k datu 24. 4. 2014.

6.1 Počet instalací

Aplikace je aktuálně nainstalována na 185 aktivních zařízeních a celkově byla nainstalována na 643 zařízeních obrázek 22.

Stránka 1 z celkového počtu 1

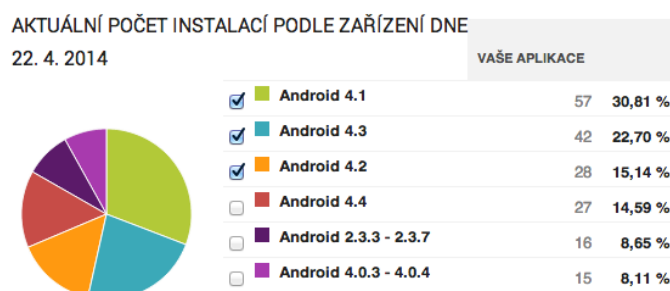
NÁZEV APLIKACE	CENA	AKTUÁLNÍ/CELKOVÝ POČET INSTALACÍ	PRŮM. HODN. / CELKEM	SELHÁNÍ A CHYBY ANR	POSLEDNÍ AKTUALIZACE	STAV
 TopNET live Mobile 3.1.0	Zdarma	185 / 643	★ 4,25 / 8	3	14. 4. 2014	Publikováno

Stránka 1 z celkového počtu 1

Obrázek 22: TopNET Live Mobile na Google play

6.2 Verze systému

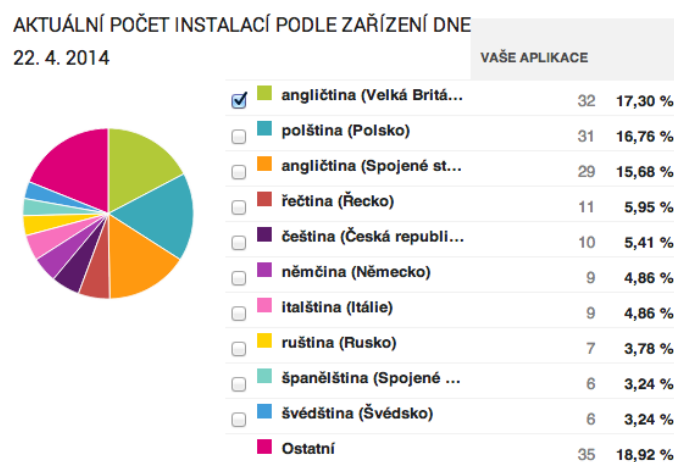
Poměrně zajímavou statistikou je zobrazení zastoupení jednotlivých verzí systému Android, na kterých je aplikace nainstalována obrázek 23. Z této statistiky je zřejmé, že uživatelé této aplikace používají hlavně novější verze systému Android. Verzi 4.0 a vyšší používá kolem 90 % uživatelů.



Obrázek 23: Aktuální počet instalací podle verze Android systému

6.3 Jazyková verze

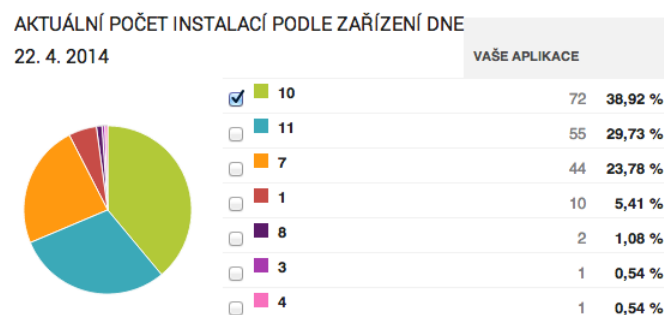
Podle této statistiky můžeme posoudit, v jakých jazykových mutacích by mělo smysl mít aplikaci. Tato aplikace je pouze v angličtině, což odpovídá, nejvíce uživatelů používá angličtinu, ale ze statistik je vidět, že poměrně velká část uživatelů také používá polštinu obrázek 24.



Obrázek 24: Aktuální počet instalací podle jazyka uživatele

6.4 Verze aplikace

Pro vývojáře, je zajímavá informace o tom kolik uživatelů si již provedlo aktualizaci aplikace a kolik ještě pracuje se starší verzí. Podle této statistiky poslední verzi 11 (interní číslování, odpovídá verzi 3.1.0) má nainstalováno 55 uživatelů obrázek 25.



Obrázek 25: Aktuální počet instalací podle verze aplikace

6.5 Selhání aplikace a chyby ANR

Pokud uživateli aplikace spadne nebo neodpovídá, má možnost poslat tuto informaci včetně výpisu zásobníku při pádu vývojáři na Google play. Vývojář dostane tuto informaci z vývojářské konzole obrázek 26. V tomto případě vidíme chybu `NullPointerException` v `MainActivity` v metodě `onSignUp`. Je možné nechat si vypsat kompletní stav zásobníků a také zjistit na jakém typu zařízení chyba nastala.

SELHÁNÍ A CHYBY ANR

Typ	Ukázat skryté	Naposledy nahlášeno	Verze systému Android	Verze aplikace	Zařízení
Selhání	ANO	Posledních 30 dní	Všechny verze	Všechny verze	Přidat filtr

1 nové selhání | 1 selhání celkem

Stránka 1 z celkového počtu 1

NÁZEV	NOVÁ	PŘEHLEDY TENTO TÝDEN	CELKEM V PŘEHLEDECH	NAPOSLEDY NAHLÁŠENO	SKRÝT
<code>java.lang.NullPointerException</code> Kde: <code>navmatix.android.tlmmobile.view.MainActivity.onSignUp</code>	NOVÁ	1	1	Včera, 8:36	Skrýt

Stránka 1 z celkového počtu 1

Obrázek 26: Selhání aplikace a chyby ANR

7 Závěr

Aplikace se vyvíjela, protože na trhu není k dispozici podobný produkt pro systém Android a také proto, že každý provozovatel sítě referenčních stanic implementuje vlastní rozhraní pro připojení k síti pro účely monitorování stavu sítě (Webové služby), takže aplikace se musela navrhnout pro konkrétní síť a její přístupové rozhraní.

Ostatní firmy používají hlavně desktopové aplikace na PC pro monitoring sítě. Od firmy Trimble je k dispozici jediná aplikace Trimble RTX, která je trochu jinak zaměřená, umí se spojit s gps přijímačem a přijímat korekce. Jedinou podobnou funkcionalitu má v zobrazení skyplotu.

Aplikaci TopNET Live v současné době používá 185 uživatelů a celkem ji mělo nainstalováno přes 600 uživatelů. Je to relativně málo, ale tato aplikace je hodně specifická pro poměrně úzký okruh uživatelů.

Zpětnou vazbu od uživatelů, dostáváme ze dvou zdrojů. V rámci Google play mají uživatelé možnost ohodnotit aplikaci a přidat případně komentář. V současné době ohodnotilo aplikaci 8 uživatelů a dostala hodnocení 4.25 z možných 5 hvězdiček obrázek 22.

Druhý zdroj zpětné vazby je přímo z firmy TopCon pro kterou se aplikace vyvíjela a jejich uživatelé jsou z aplikaci spokojeni, jak nám potvrdilo vedení firmy a objednalo si vytvoření stejné aplikace i pro operační systém iOS od firmy Apple. Firma TopCon má hodně uživatelů s telefony IPHONE a těm se líbila stávající aplikace pro Android.

Na základě připomínek uživatelů se připravuje rozšíření aplikace o další modul, který bude používat Google Cloud Messaging pro odeslání upozornění do telefonu při monitoringu sítě.

8 Reference

- [1] J. Friesen: Learn Java for Android Development, 2013, Apress, ISBN: 978-1430257226
- [2] Z. Mednieks: Programming Android: Java Programming for the New Generation of Mobile Devices, 2012, O'Reilly Media, ISBN: 978-1449316648
- [3] NOLAN, Godfrey. Android Best Practices. New edition. Apress, 2013. ISBN 978-143-0258-575.
- [4] Android recipes: a problem-solution approach. Berkeley: Apress, 2014. ISBN 978-143-0263-227.
- [5] RAPANT, Petr. Geoinformatika a geoinformační technologie. 1. vyd. Ostrava: VŠB - Technická univerzita Ostrava, Hornicko-geologická fakulta, Institut geoinformatiky, 2006, xxxv, 463 s. ISBN 80-248-1264-9.
- [6] [online]. [cit. 2014-04-02]. Dostupné z: <http://developer.android.com>
- [7] [online]. [cit. 2014-04-02]. Dostupné z: <http://en.wikipedia.org/wiki/Google-Play>
- [8] [online]. [cit. 2014-04-08]. Dostupné z: <http://cs.wikipedia.org/wiki/Android>
- [9] [online]. [cit. 2014-04-10]. Dostupné z: <http://developer.android.com/tools/index.html>
- [10] [online]. [cit. 2014-04-08]. Dostupné z: <http://www.can-net.ca/cannet/index.php>
- [11] [online]. [cit. 2014-04-08]. Dostupné z: <http://www.beruna.cz/text-metody-kterymlze-z-gps-dostat-milimetry>
- [12] [online]. [cit. 2014-04-08]. Dostupné z: <http://cs.wikipedia.org/wiki/Globální-družicový-polohový-systém>
- [13] [online]. [cit. 2014-04-08]. Dostupné z: <http://http://topnet.geodis.cz/topnet/topnet.aspx>